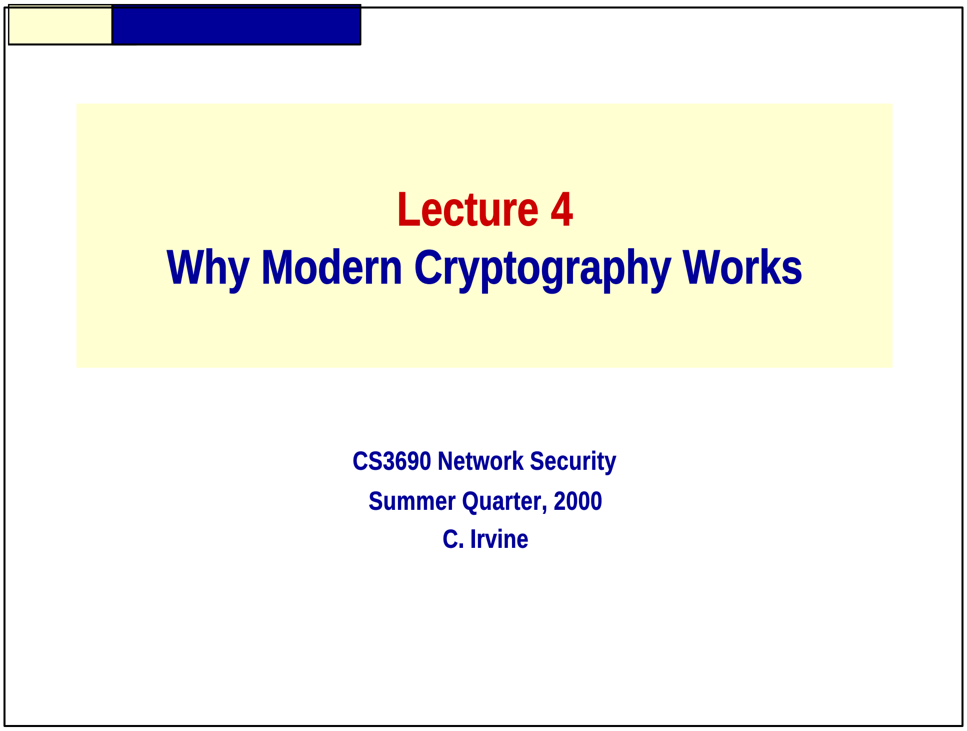
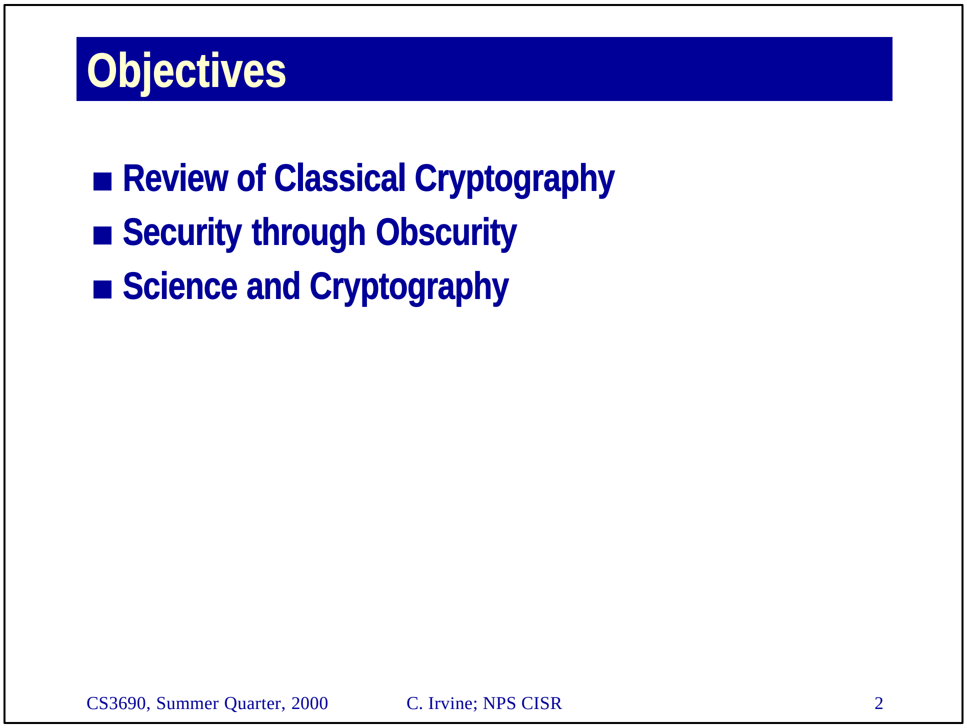


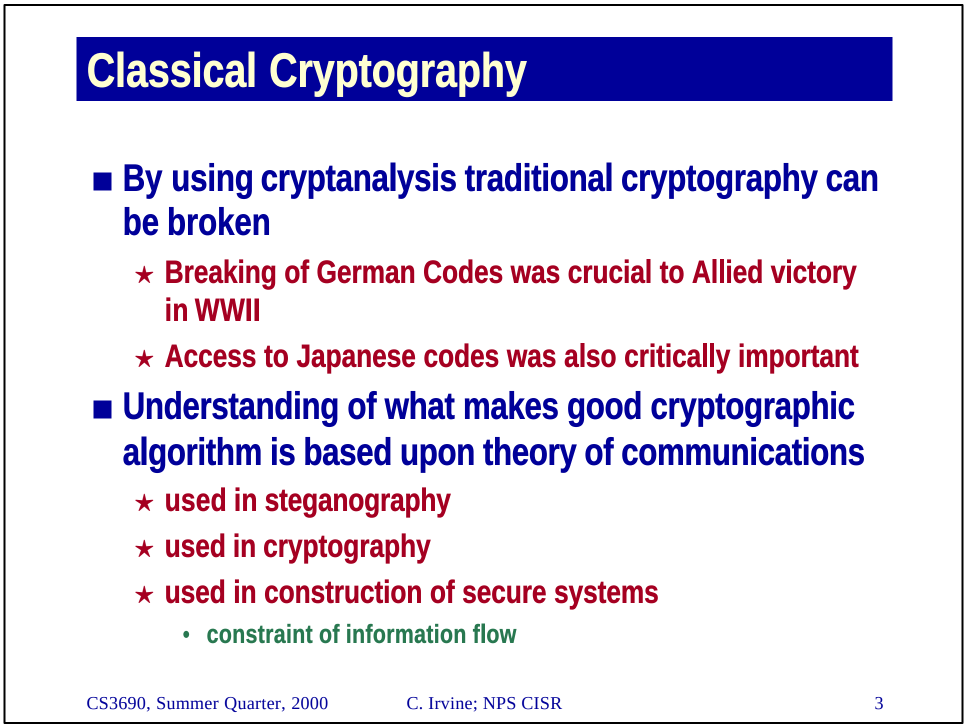
Lecture 4
Why Modern Cryptography Works

CS3690 Network Security
Summer Quarter, 2000
C. Irvine



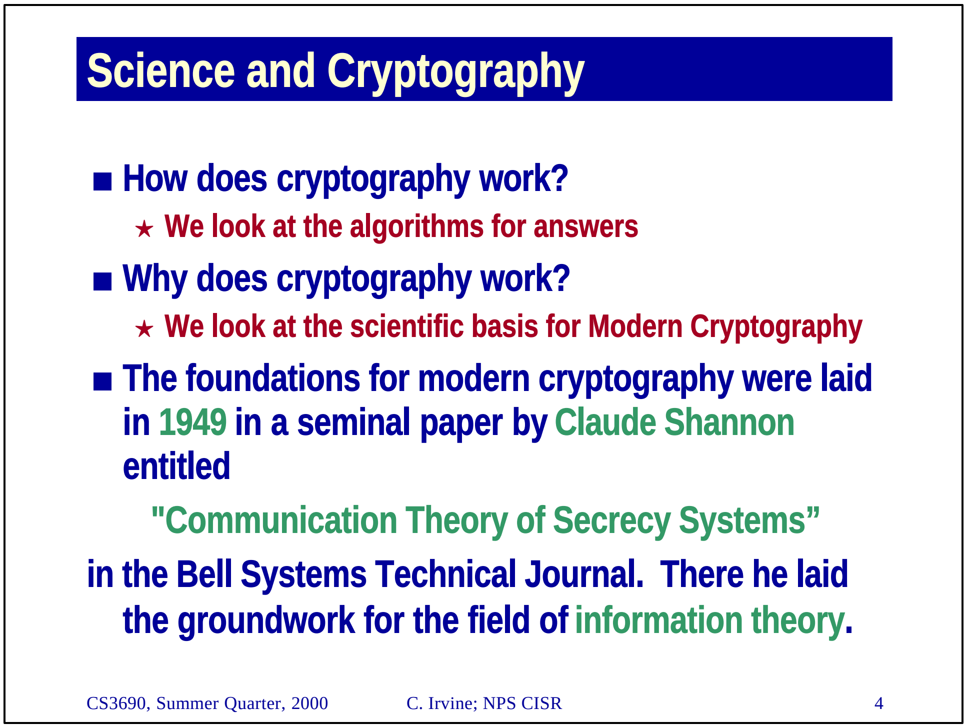
Objectives

- Review of Classical Cryptography
- Security through Obscurity
- Science and Cryptography



Classical Cryptography

- By using cryptanalysis traditional cryptography can be broken
  - ★ Breaking of German Codes was crucial to Allied victory in WWII
  - ★ Access to Japanese codes was also critically important
- Understanding of what makes good cryptographic algorithm is based upon theory of communications
  - ★ used in steganography
  - ★ used in cryptography
  - ★ used in construction of secure systems
    - constraint of information flow

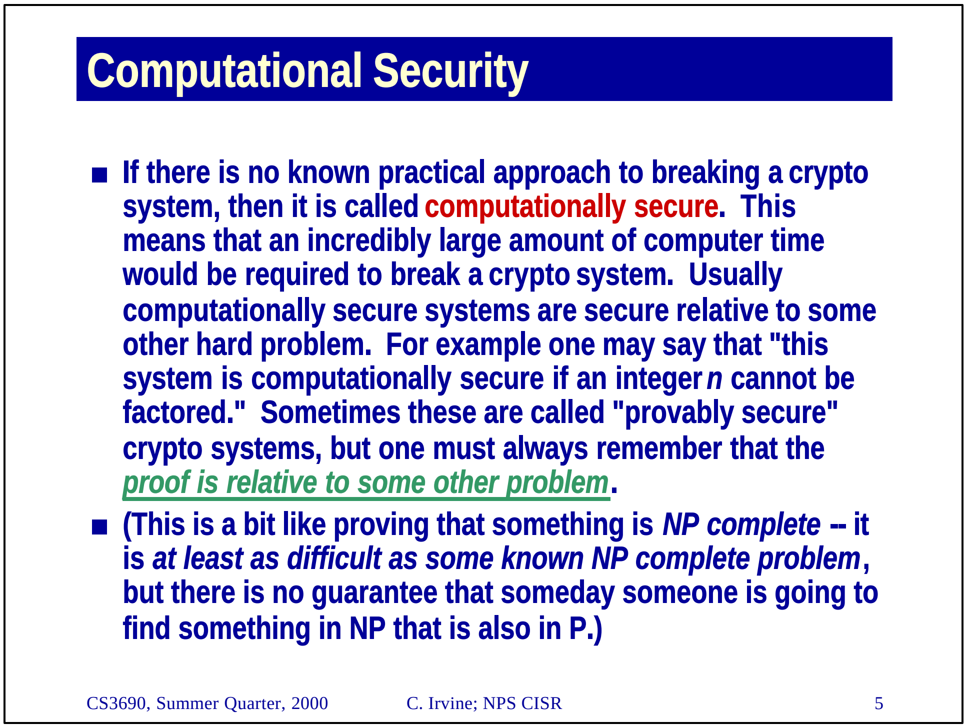


Science and Cryptography

- How does cryptography work?
  - ★ We look at the algorithms for answers
- Why does cryptography work?
  - ★ We look at the scientific basis for Modern Cryptography
- The foundations for modern cryptography were laid in 1949 in a seminal paper by Claude Shannon entitled

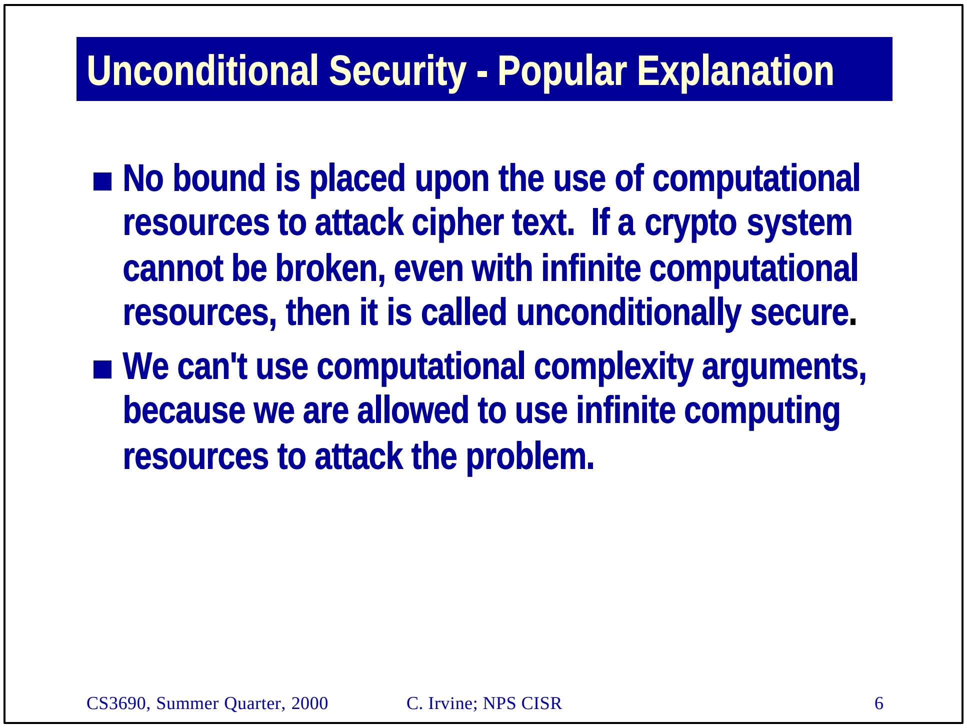
"Communication Theory of Secrecy Systems"

in the Bell Systems Technical Journal. There he laid the groundwork for the field of information theory.



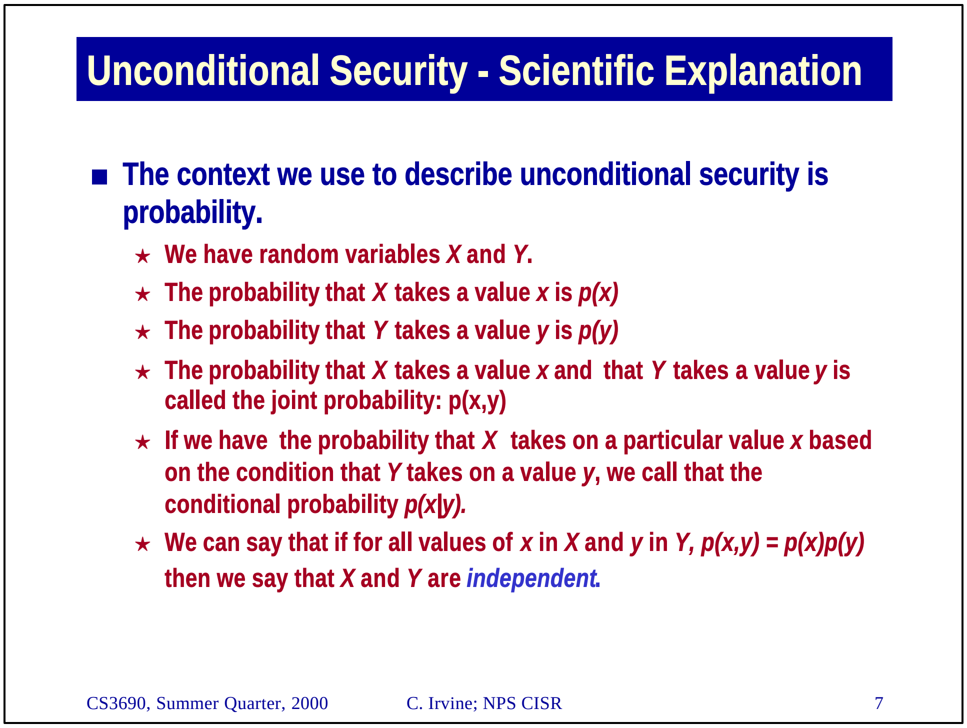
Computational Security

- If there is no known practical approach to breaking a crypto system, then it is called computationally secure. This means that an incredibly large amount of computer time would be required to break a crypto system. Usually computationally secure systems are secure relative to some other hard problem. For example one may say that "this system is computationally secure if an integer *n* cannot be factored." Sometimes these are called "provably secure" crypto systems, but one must always remember that the *proof is relative to some other problem*.
- (This is a bit like proving that something is *NP complete* -- it is *at least as difficult as some known NP complete problem*, but there is no guarantee that someday someone is going to find something in NP that is also in P.)



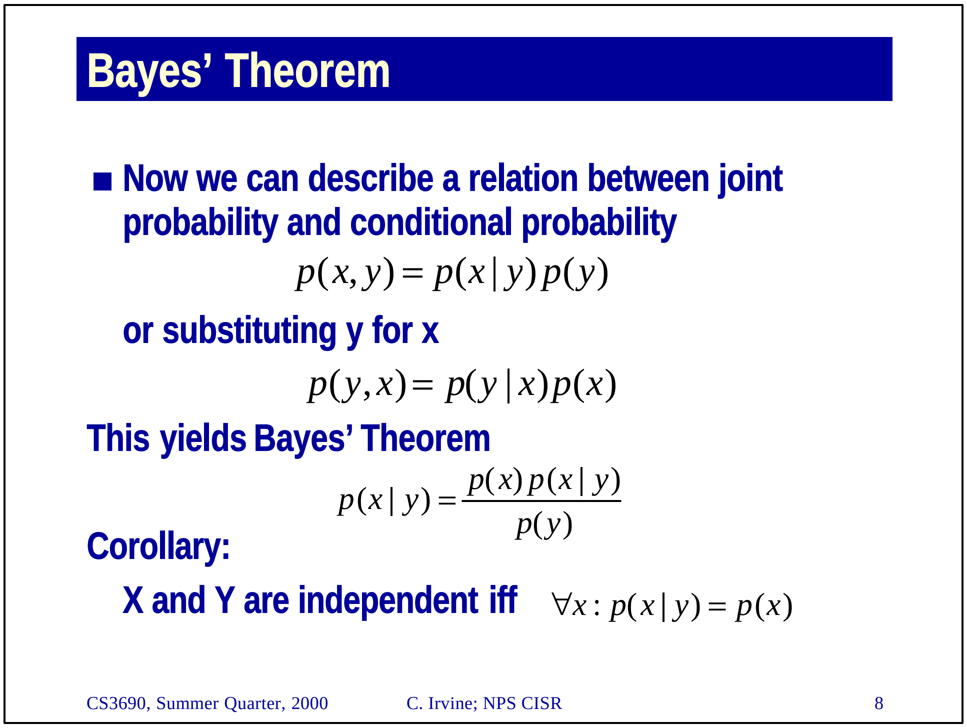
Unconditional Security - Popular Explanation

- No bound is placed upon the use of computational resources to attack cipher text. If a crypto system cannot be broken, even with infinite computational resources, then it is called unconditionally secure.
- We can't use computational complexity arguments, because we are allowed to use infinite computing resources to attack the problem.



Unconditional Security - Scientific Explanation

- The context we use to describe unconditional security is probability.
  - We have random variables *X* and *Y*.
  - The probability that *X* takes a value *x* is *p(x)*
  - The probability that *Y* takes a value *y* is *p(y)*
  - The probability that *X* takes a value *x* and that *Y* takes a value *y* is called the joint probability: p(x,y)
  - If we have the probability that *X* takes on a particular value *x* based on the condition that *Y* takes on a value *y*, we call that the conditional probability *p(x|y).*
  - We can say that if for all values of *x* in *X* and *y* in *Y*, *p(x,y) = p(x)p(y)* then we say that *X* and *Y* are ***independent.***



Bayes' Theorem

- Now we can describe a relation between joint probability and conditional probability

$$p(x,y) = p(x|y)p(y)$$

or substituting y for x

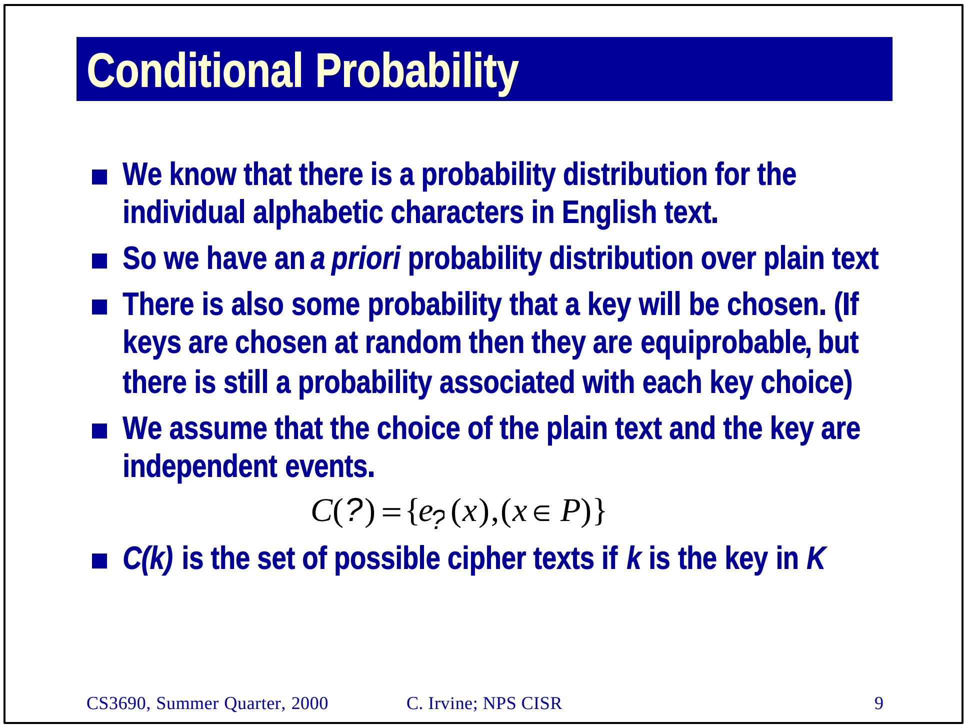
$$p(y,x) = p(y|x)p(x)$$

This yields Bayes' Theorem

$$p(x|y) = \frac{p(x)p(x|y)}{p(y)}$$

Corollary:

X and Y are independent iff $\forall x : p(x|y) = p(x)$

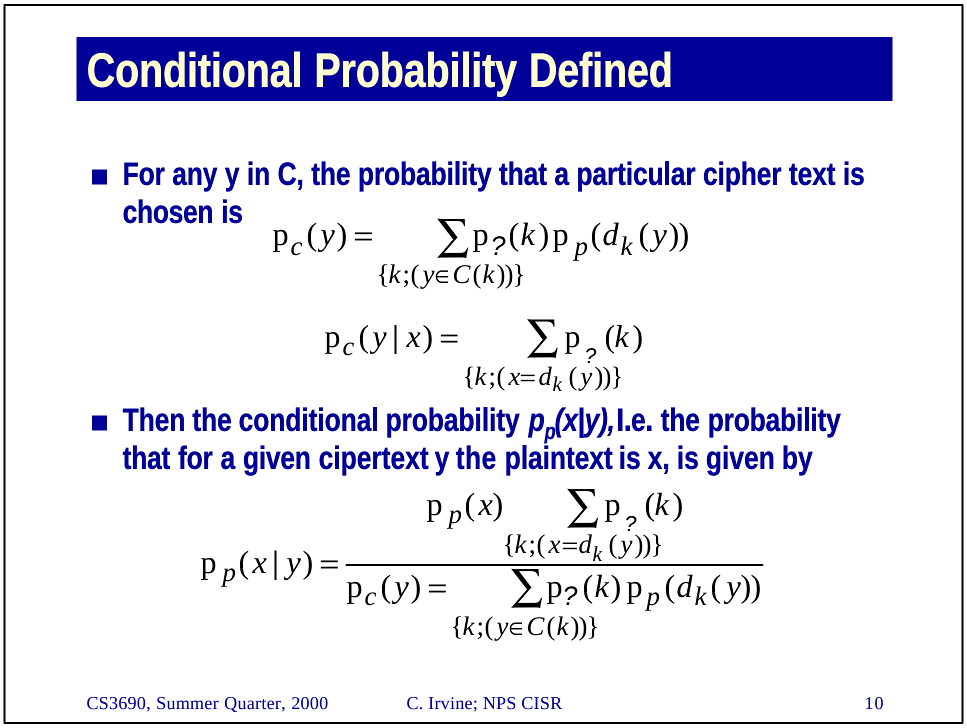


Conditional Probability

- We know that there is a probability distribution for the individual alphabetic characters in English text.
- So we have an *a priori* probability distribution over plain text
- There is also some probability that a key will be chosen. (If keys are chosen at random then they are equiprobable, but there is still a probability associated with each key choice)
- We assume that the choice of the plain text and the key are independent events.

$$C(?) = \{e_?(x), (x \in P)\}$$

- *C(k)* is the set of possible cipher texts if *k* is the key in *K*



Conditional Probability Defined

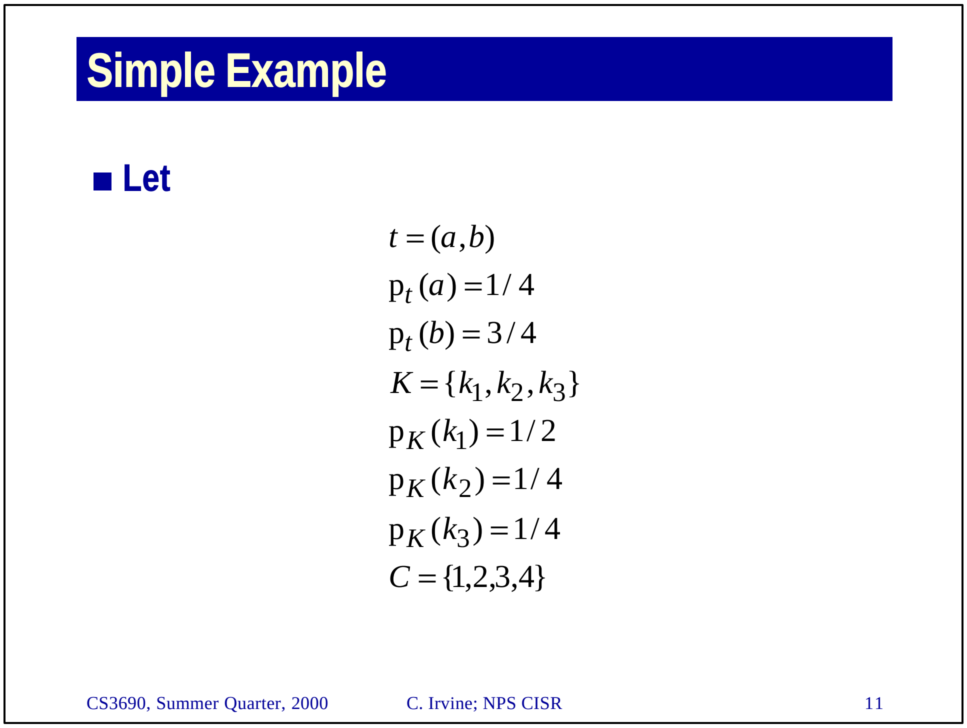
- For any y in C, the probability that a particular cipher text is chosen is

$$p_c(y) = \sum_{\{k;(y \in C(k))\}} p_?(k) p_p(d_k(y))$$

$$p_c(y|x) = \sum_{\{k;(x=d_k(y))\}} p_?(k)$$

- Then the conditional probability *$p_p(x|y)$*, I.e. the probability that for a given cipertext y the plaintext is x, is given by

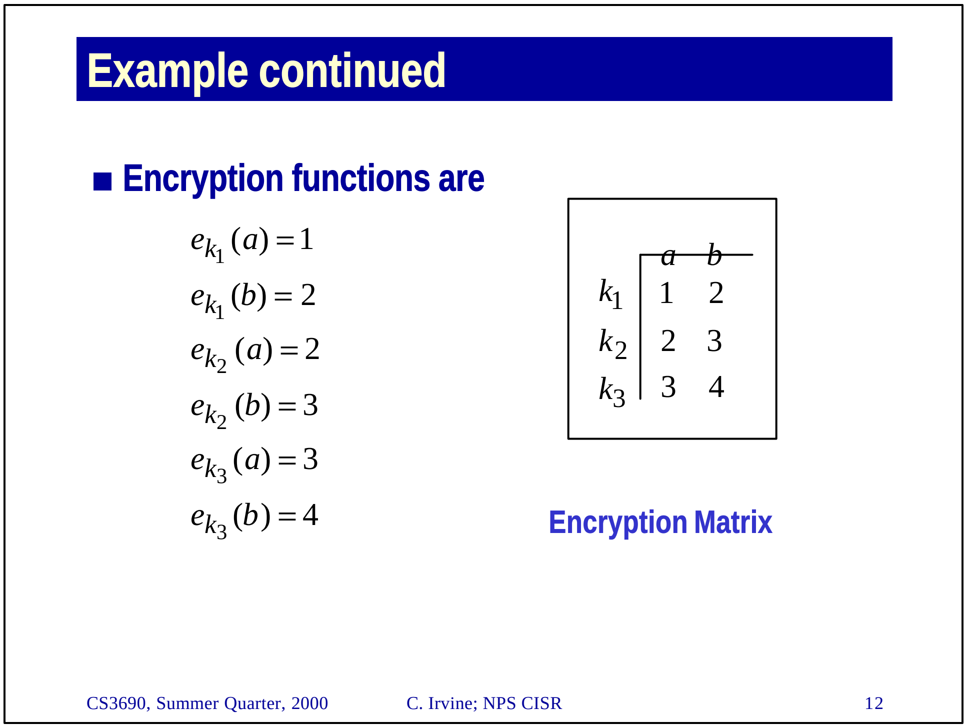
$$p_p(x|y) = \frac{p_p(x) \sum_{\{k;(x=d_k(y))\}} p_?(k)}{p_c(y) = \sum_{\{k;(y \in C(k))\}} p_?(k) p_p(d_k(y))}$$



Simple Example

- Let

$$
\begin{aligned}
&t = (a, b)\\
&p_t(a) = 1/4\\
&p_t(b) = 3/4\\
&K = \{k_1, k_2, k_3\}\\
&p_K(k_1) = 1/2\\
&p_K(k_2) = 1/4\\
&p_K(k_3) = 1/4\\
&C = \{1,2,3,4\}
\end{aligned}
$$



Example continued

- Encryption functions are

$$
\begin{aligned}
&e_{k_1}(a) = 1\\
&e_{k_1}(b) = 2\\
&e_{k_2}(a) = 2\\
&e_{k_2}(b) = 3\\
&e_{k_3}(a) = 3\\
&e_{k_3}(b) = 4
\end{aligned}
$$

$$
\begin{array}{c|cc}
 & a & b \\
\hline
k_1 & 1 & 2 \\
k_2 & 2 & 3 \\
k_3 & 3 & 4
\end{array}
$$

Encryption Matrix

Ciphertext Probability Distribution

- Recall there are four cipher characters: 1, 2, 3, 4
- We have to compute the probability of the cipher character appearing as a result of all possible combinations of plain text and keys
- Consider cipher character 1. It can only be produced using plaintext character a and key 1

$$p_c(1) = \tfrac{1}{8}$$

$$p_c(2) = \tfrac{3}{8} + \tfrac{1}{16} = \tfrac{7}{16}$$

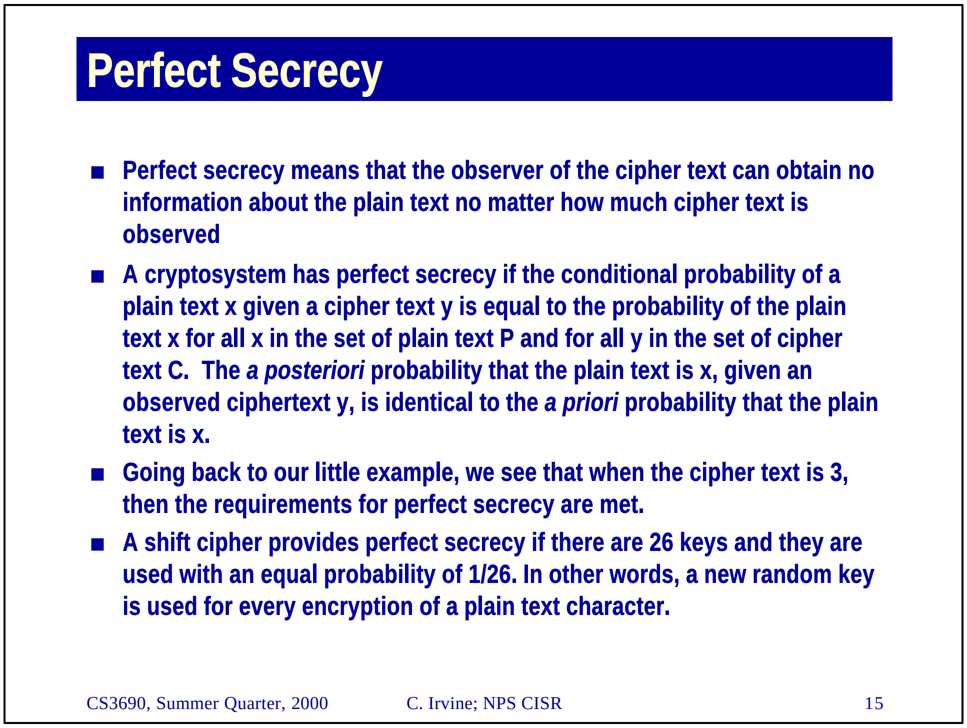
$$p_c(2) = \tfrac{3}{16} + \tfrac{1}{16} = \tfrac{1}{4}$$

$$p_c(4) = \tfrac{3}{8}$$

Conditional Probability Distributions

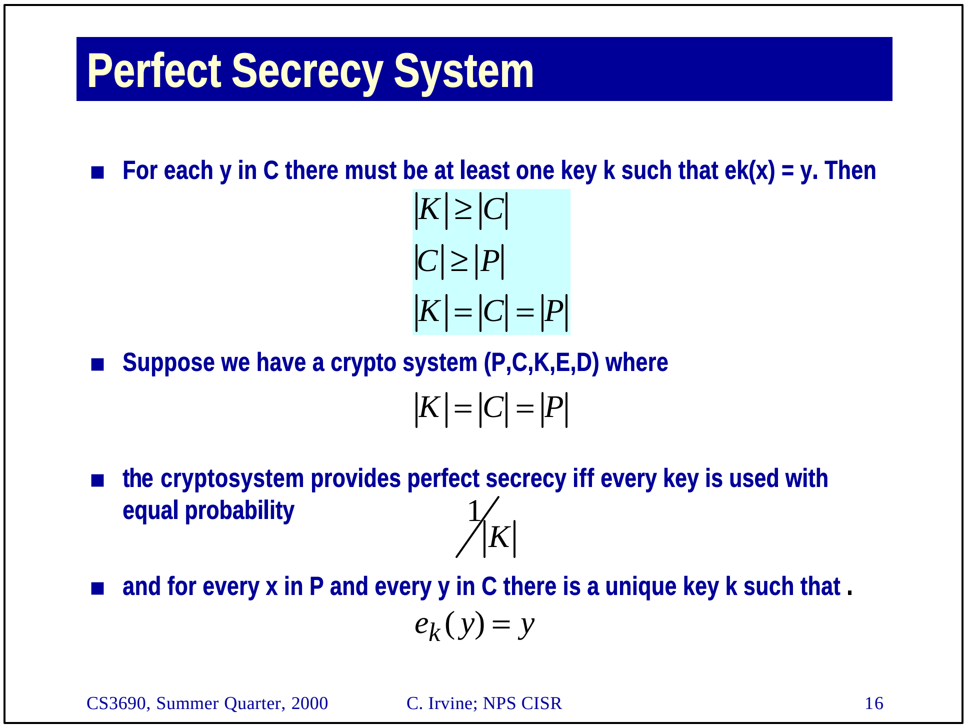
- We want the probability of plain text given some observed cipher text. Using our example:

$$\begin{aligned}
\mathrm{p}_p(a|1) &= 1 & \qquad \mathrm{p}_p(b|1) &= 0 \\
\mathrm{p}_p(a|2) &= \tfrac{1}{7} & \qquad \mathrm{p}_p(b|2) &= \tfrac{6}{7} \\
\mathrm{p}_p(a|3) &= \tfrac{1}{4} & \qquad \mathrm{p}_p(b|3) &= \tfrac{3}{4} \\
\mathrm{p}_p(a|4) &= 0 & \qquad \mathrm{p}_p(b|4) &= 1
\end{aligned}$$



Perfect Secrecy

- Perfect secrecy means that the observer of the cipher text can obtain no information about the plain text no matter how much cipher text is observed
- A cryptosystem has perfect secrecy if the conditional probability of a plain text x given a cipher text y is equal to the probability of the plain text x for all x in the set of plain text P and for all y in the set of cipher text C. The *a posteriori* probability that the plain text is x, given an observed ciphertext y, is identical to the *a priori* probability that the plain text is x.
- Going back to our little example, we see that when the cipher text is 3, then the requirements for perfect secrecy are met.
- A shift cipher provides perfect secrecy if there are 26 keys and they are used with an equal probability of 1/26. In other words, a new random key is used for every encryption of a plain text character.



Perfect Secrecy System

- For each y in C there must be at least one key k such that ek(x) = y. Then

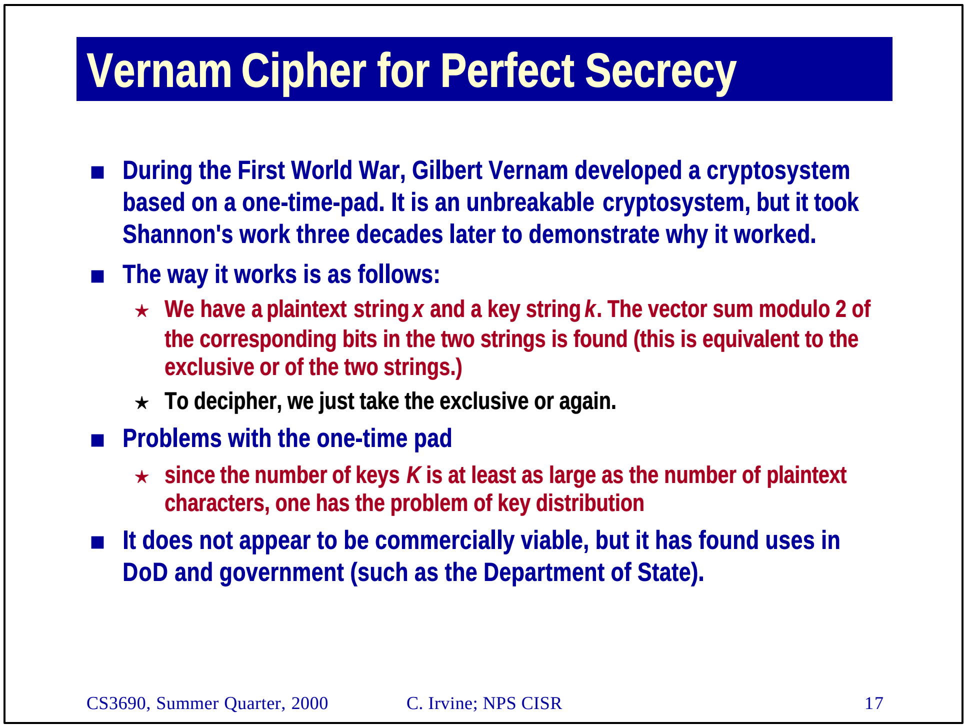
$$|K| \ge |C|$$
$$|C| \ge |P|$$
$$|K| = |C| = |P|$$

- Suppose we have a crypto system (P,C,K,E,D) where

$$|K| = |C| = |P|$$

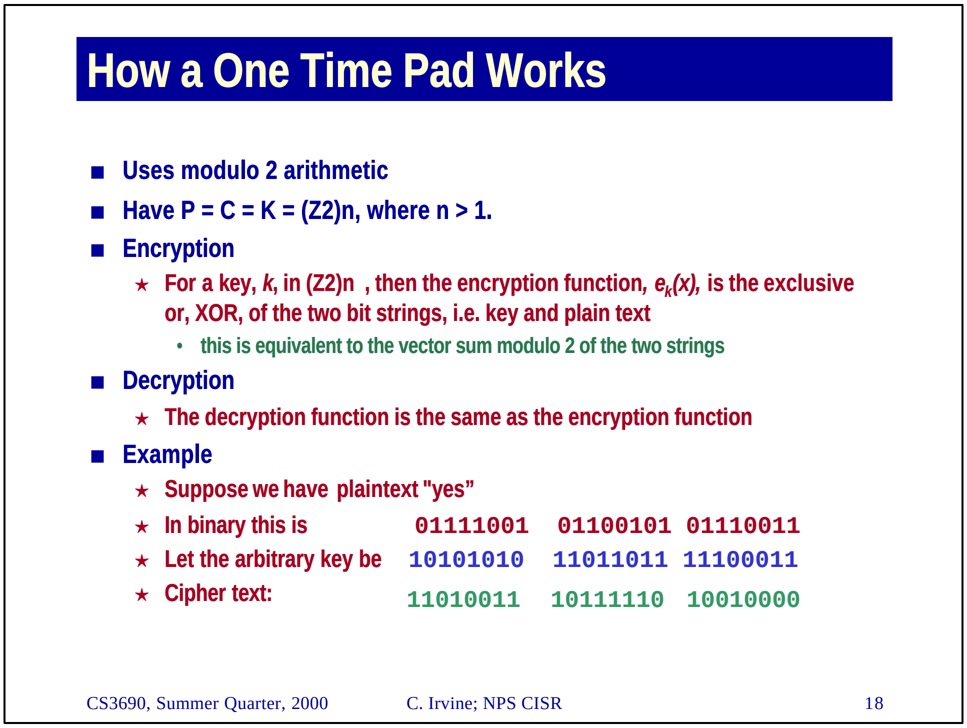
- the cryptosystem provides perfect secrecy iff every key is used with equal probability $1/|K|$
- and for every x in P and every y in C there is a unique key k such that .

$$e_k(y) = y$$



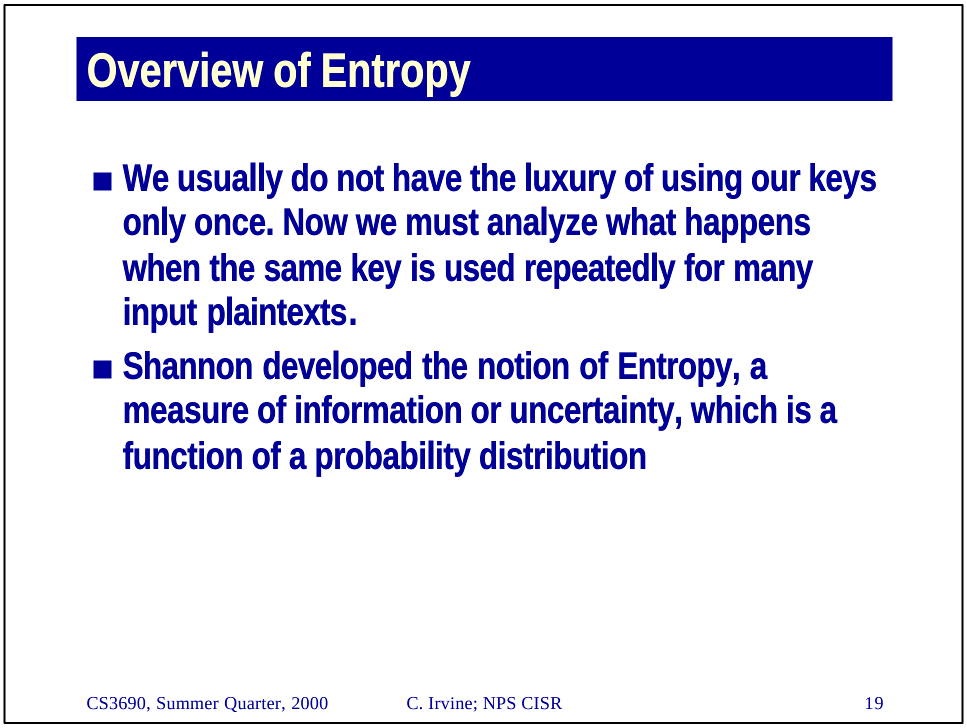
Vernam Cipher for Perfect Secrecy

- During the First World War, Gilbert Vernam developed a cryptosystem based on a one-time-pad. It is an unbreakable cryptosystem, but it took Shannon's work three decades later to demonstrate why it worked.
- The way it works is as follows:
  - ★ We have a plaintext string *x* and a key string *k*. The vector sum modulo 2 of the corresponding bits in the two strings is found (this is equivalent to the exclusive or of the two strings.)
  - ★ To decipher, we just take the exclusive or again.
- Problems with the one-time pad
  - ★ since the number of keys *K* is at least as large as the number of plaintext characters, one has the problem of key distribution
- It does not appear to be commercially viable, but it has found uses in DoD and government (such as the Department of State).



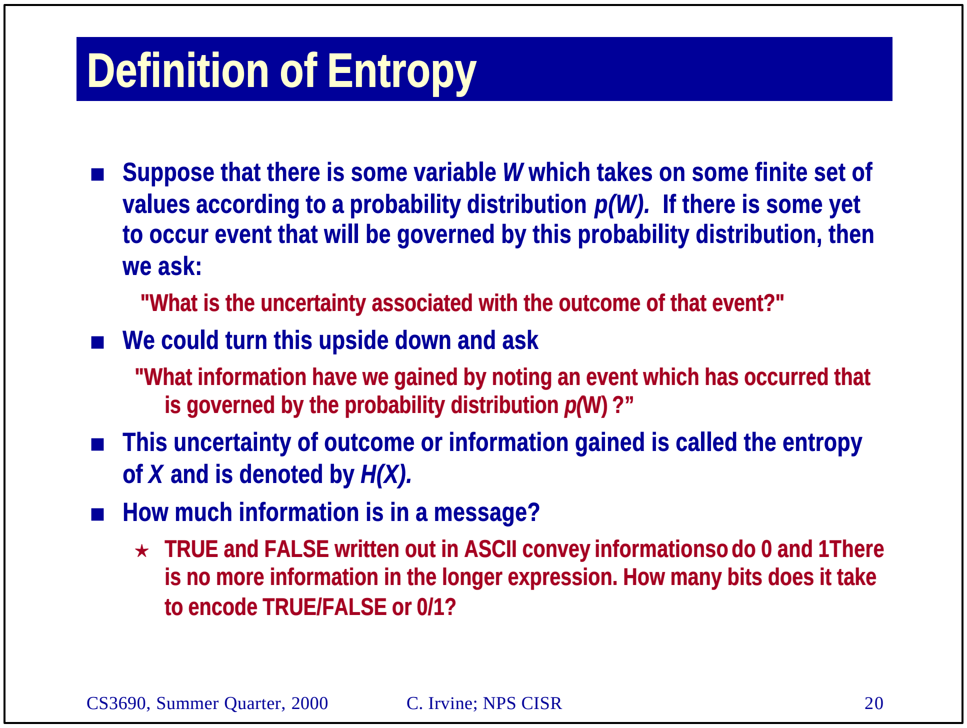
How a One Time Pad Works

- Uses modulo 2 arithmetic
- Have P = C = K = (Z2)n, where n > 1.
- Encryption
  - ★ For a key, *k*, in (Z2)n , then the encryption function, $e_k(x)$, is the exclusive or, XOR, of the two bit strings, i.e. key and plain text
    - this is equivalent to the vector sum modulo 2 of the two strings
- Decryption
  - ★ The decryption function is the same as the encryption function
- Example
  - ★ Suppose we have plaintext "yes"
  - ★ In binary this is 01111001 01100101 01110011
  - ★ Let the arbitrary key be 10101010 11011011 11100011
  - ★ Cipher text: 11010011 10111110 10010000



Overview of Entropy

- We usually do not have the luxury of using our keys only once. Now we must analyze what happens when the same key is used repeatedly for many input plaintexts.
- Shannon developed the notion of Entropy, a measure of information or uncertainty, which is a function of a probability distribution

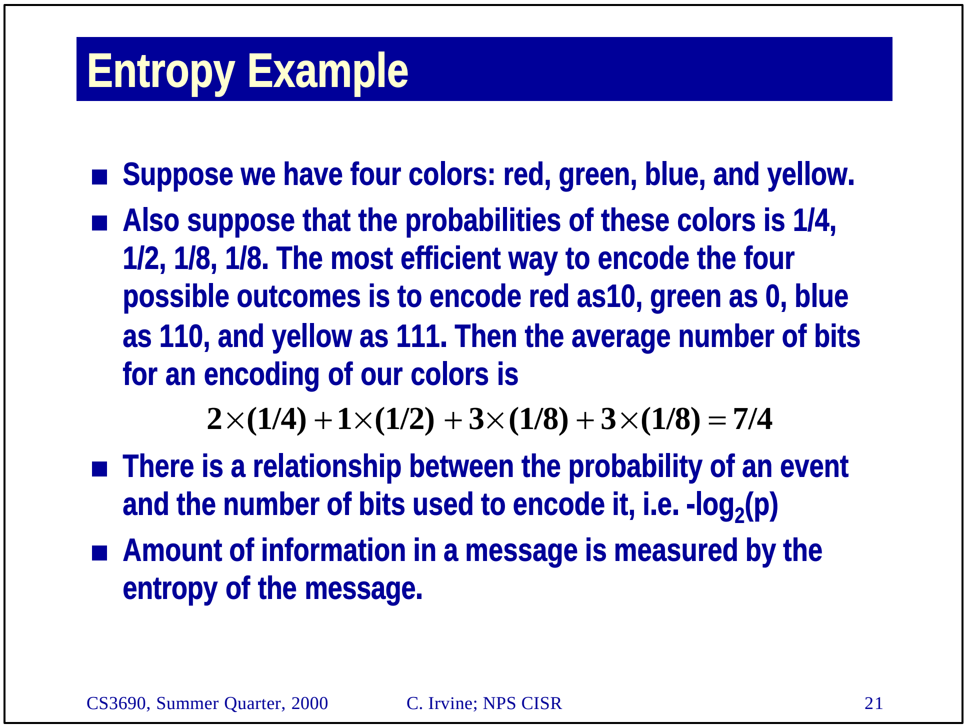


Definition of Entropy

- Suppose that there is some variable *W* which takes on some finite set of values according to a probability distribution *p(W).* If there is some yet to occur event that will be governed by this probability distribution, then we ask:

  "What is the uncertainty associated with the outcome of that event?"
- We could turn this upside down and ask

  "What information have we gained by noting an event which has occurred that is governed by the probability distribution *p*(W) ?”
- This uncertainty of outcome or information gained is called the entropy of *X* and is denoted by *H(X).*
- How much information is in a message?
  - ★ TRUE and FALSE written out in ASCII convey informationso do 0 and 1There is no more information in the longer expression. How many bits does it take to encode TRUE/FALSE or 0/1?

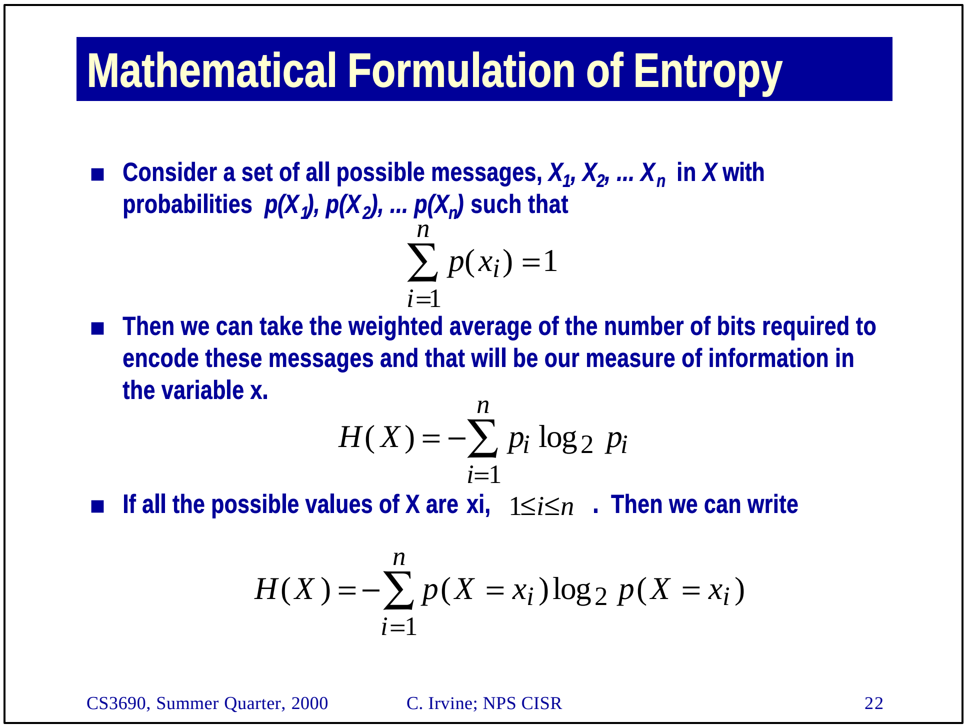


Entropy Example

- Suppose we have four colors: red, green, blue, and yellow.
- Also suppose that the probabilities of these colors is 1/4, 1/2, 1/8, 1/8. The most efficient way to encode the four possible outcomes is to encode red as10, green as 0, blue as 110, and yellow as 111. Then the average number of bits for an encoding of our colors is

$$2\times(1/4)+1\times(1/2)+3\times(1/8)+3\times(1/8)=7/4$$

- There is a relationship between the probability of an event and the number of bits used to encode it, i.e. $-\log_2(p)$
- Amount of information in a message is measured by the entropy of the message.



Mathematical Formulation of Entropy

- Consider a set of all possible messages, $X_1, X_2, \ldots X_n$ in X with probabilities $p(X_1), p(X_2), \ldots p(X_n)$ such that

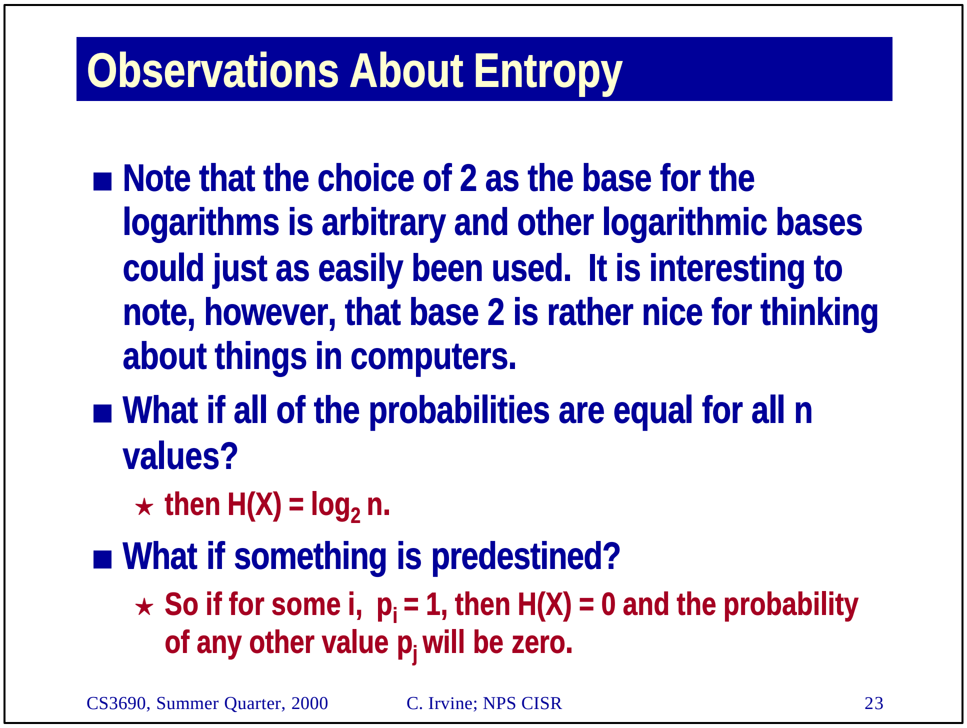
$$\sum_{i=1}^{n} p(x_i) = 1$$

- Then we can take the weighted average of the number of bits required to encode these messages and that will be our measure of information in the variable x.

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

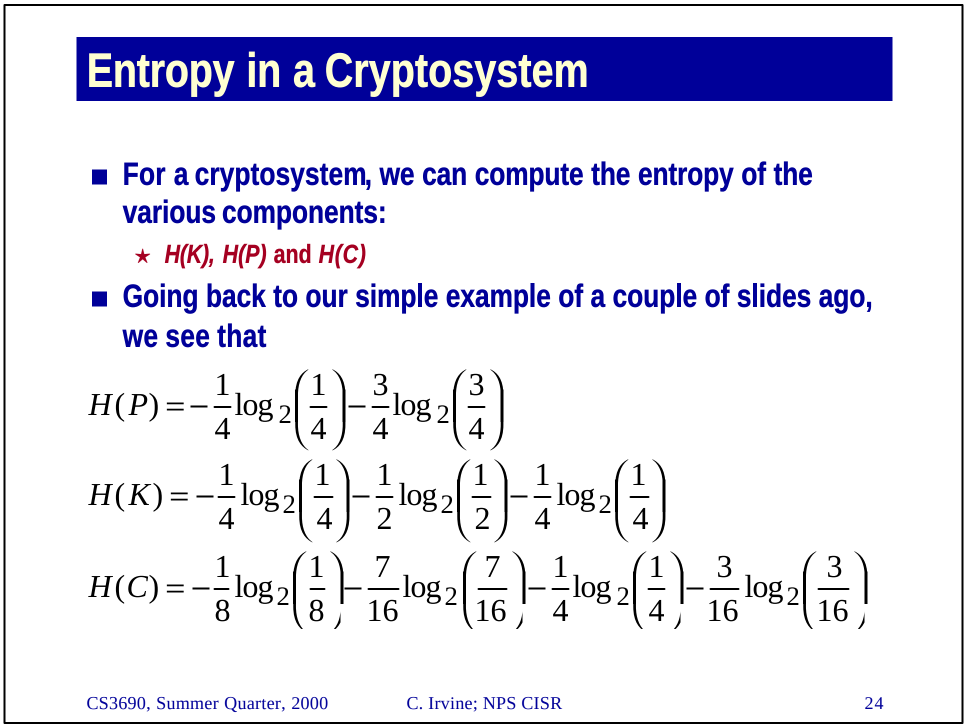
- If all the possible values of X are xi, $1 \le i \le n$. Then we can write

$$H(X) = -\sum_{i=1}^{n} p(X = x_i) \log_2 p(X = x_i)$$



Observations About Entropy

- Note that the choice of 2 as the base for the logarithms is arbitrary and other logarithmic bases could just as easily been used. It is interesting to note, however, that base 2 is rather nice for thinking about things in computers.
- What if all of the probabilities are equal for all n values?
  - ★ then $H(X) = \log_2 n$.
- What if something is predestined?
  - ★ So if for some i, $p_i = 1$, then $H(X) = 0$ and the probability of any other value p_j will be zero.



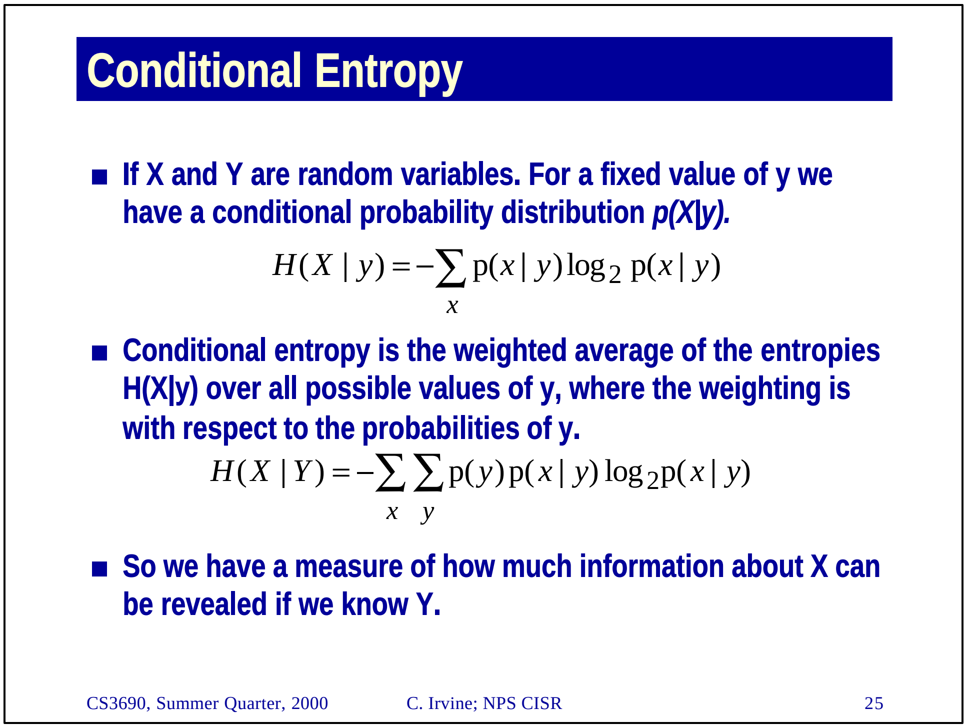
Entropy in a Cryptosystem

- For a cryptosystem, we can compute the entropy of the various components:
  - ★ *H(K), H(P)* and *H(C)*
- Going back to our simple example of a couple of slides ago, we see that

$$H(P) = -\frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{3}{4}\log_2\left(\frac{3}{4}\right)$$

$$H(K) = -\frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)$$

$$H(C) = -\frac{1}{8}\log_2\left(\frac{1}{8}\right) - \frac{7}{16}\log_2\left(\frac{7}{16}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{3}{16}\log_2\left(\frac{3}{16}\right)$$



Conditional Entropy

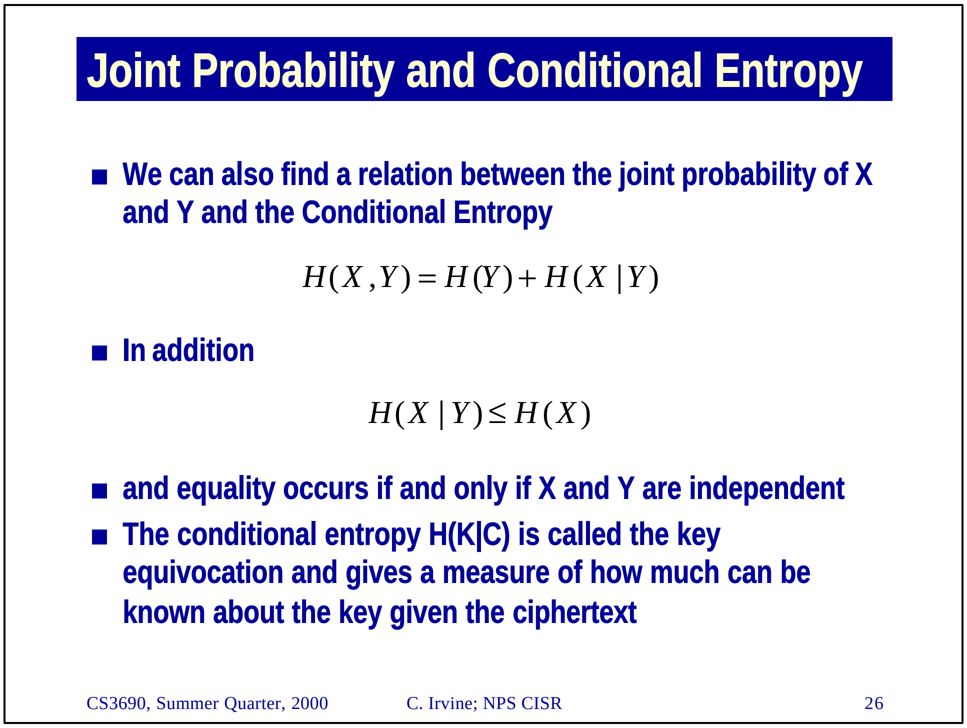
- If X and Y are random variables. For a fixed value of y we have a conditional probability distribution *p(X|y).*

$$H(X \mid y) = -\sum_{x} \mathrm{p}(x \mid y) \log_2 \mathrm{p}(x \mid y)$$

- Conditional entropy is the weighted average of the entropies H(X|y) over all possible values of y, where the weighting is with respect to the probabilities of y.

$$H(X \mid Y) = -\sum_{x} \sum_{y} \mathrm{p}(y)\, \mathrm{p}(x \mid y) \log_2 \mathrm{p}(x \mid y)$$

- So we have a measure of how much information about X can be revealed if we know Y.



Joint Probability and Conditional Entropy

- We can also find a relation between the joint probability of X and Y and the Conditional Entropy

$$H(X,Y) = H(Y) + H(X \mid Y)$$

- In addition

$$H(X \mid Y) \le H(X)$$

- and equality occurs if and only if X and Y are independent
- The conditional entropy H(K|C) is called the key equivocation and gives a measure of how much can be known about the key given the ciphertext

Applying Entropy to Cryptosystems

- Theorem: Consider the cryptosystem (P, C, K, E, D)

  $$H(K|C) = H(K) + H(P) - H(C)$$

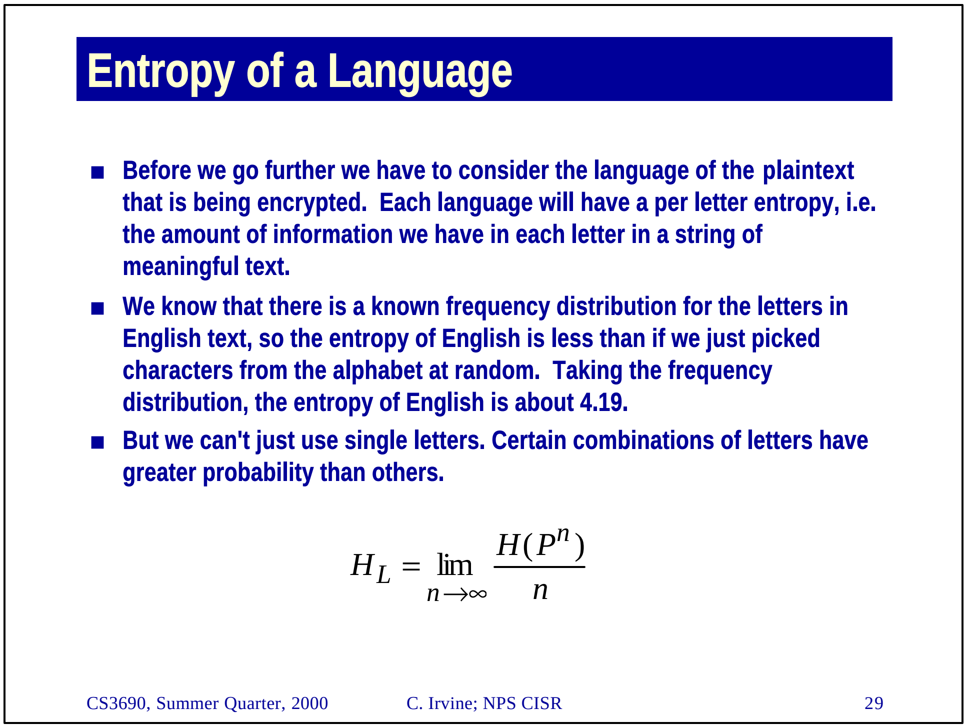
- Proof
  - ★ From the preceding slide we can see that $H(K, P, C) = H(C|K, P) + H(K,P)$
  - ★ We already know that we have an encryption algorithm so that the key and the plaintext uniquely determine the ciphertext (recall one-to-one)
    - This means that $H(C|K,P) = 0$.
  - ★ Now we have $H(K, P, C) = H(K,P)$
  - ★ We know that K and P are independent, so we can use another of our rules
    - $H(K, P, C) = H(K) + H(P)$
  - ★ Similarly, we can show that $H(K, P, C) = H(K,C)$
  - ★ So substituting

    $$\begin{aligned} H(K|C) &= H(K, C) - H(C) \\ &= H(K,P,C) - H(C) \\ &= H(K) + H(P) - H(C) \end{aligned}$$

  - ★ For the simple example we have been looking at, $H(K|C)$ is about 0.46

Spurious Keys -- A Cryptanalysis Problem

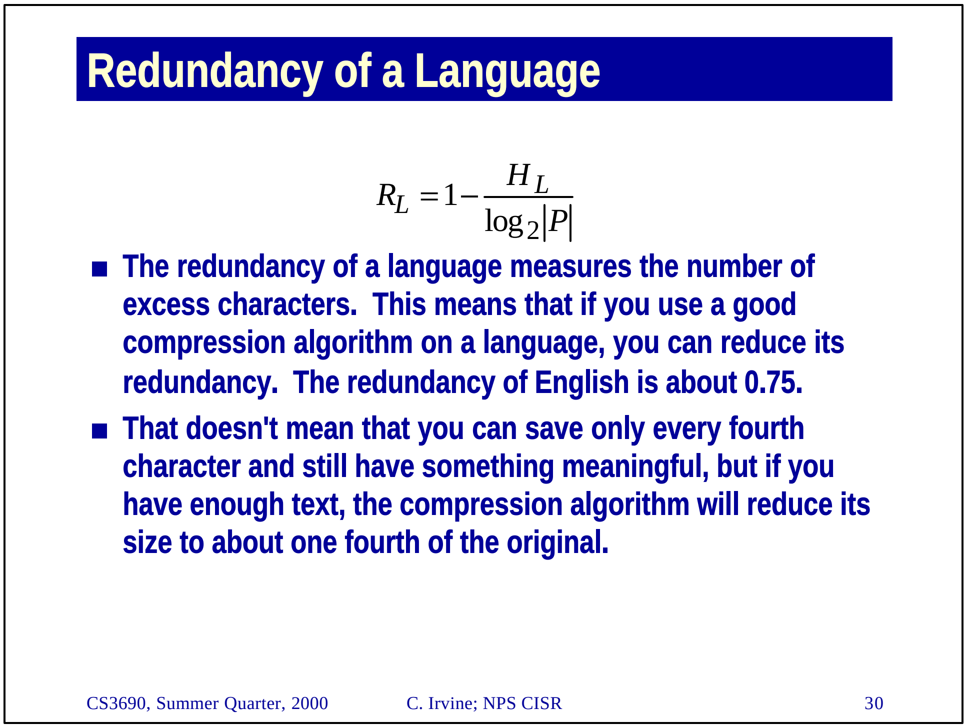
- Suppose we have a crypto system and can apply infinite computational resources to attack it be examining the cipher text.
- We can assume that the message was in some language such as English or French.
- We can rule out many keys but some keys remain, i.e. these spurious keys produce the right cipher text, but they aren't really encryptions of the correct plaintext.



Entropy of a Language

- Before we go further we have to consider the language of the plaintext that is being encrypted. Each language will have a per letter entropy, i.e. the amount of information we have in each letter in a string of meaningful text.
- We know that there is a known frequency distribution for the letters in English text, so the entropy of English is less than if we just picked characters from the alphabet at random. Taking the frequency distribution, the entropy of English is about 4.19.
- But we can't just use single letters. Certain combinations of letters have greater probability than others.

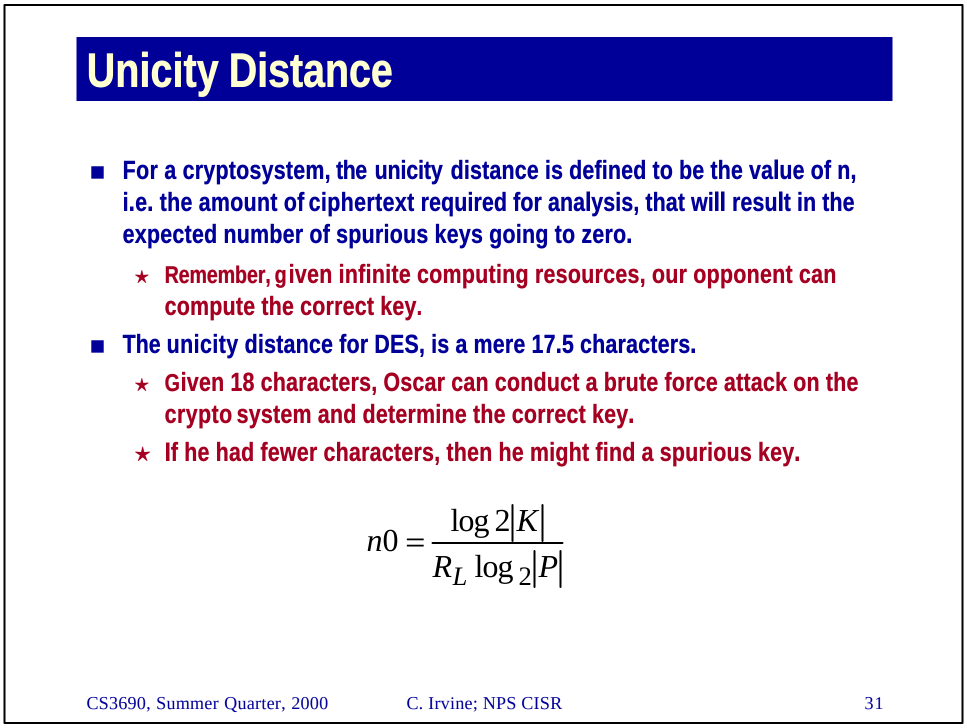
$$H_L = \lim_{n \to \infty} \frac{H(P^n)}{n}$$



Redundancy of a Language

$$R_L = 1 - \frac{H_L}{\log_2 |P|}$$

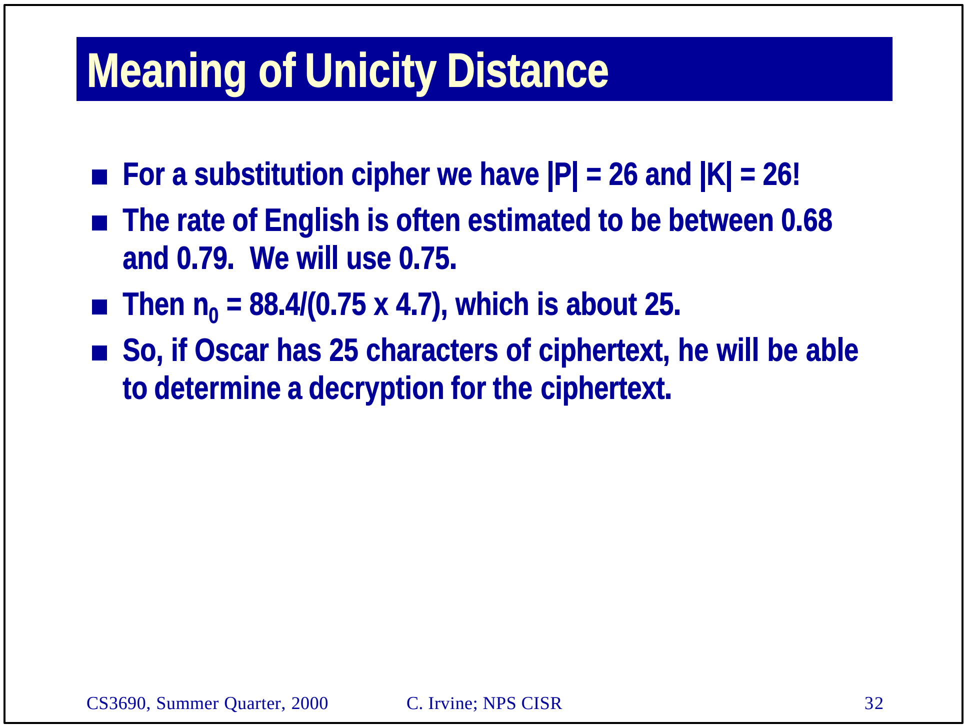
- The redundancy of a language measures the number of excess characters. This means that if you use a good compression algorithm on a language, you can reduce its redundancy. The redundancy of English is about 0.75.
- That doesn't mean that you can save only every fourth character and still have something meaningful, but if you have enough text, the compression algorithm will reduce its size to about one fourth of the original.



Unicity Distance

- For a cryptosystem, the unicity distance is defined to be the value of n, i.e. the amount of ciphertext required for analysis, that will result in the expected number of spurious keys going to zero.
  - ★ Remember, given infinite computing resources, our opponent can compute the correct key.
- The unicity distance for DES, is a mere 17.5 characters.
  - ★ Given 18 characters, Oscar can conduct a brute force attack on the crypto system and determine the correct key.
  - ★ If he had fewer characters, then he might find a spurious key.

$$n0 = \frac{\log 2|K|}{R_L \log_2|P|}$$



Meaning of Unicity Distance

- For a substitution cipher we have |P| = 26 and |K| = 26!
- The rate of English is often estimated to be between 0.68 and 0.79. We will use 0.75.
- Then n_0 = 88.4/(0.75 x 4.7), which is about 25.
- So, if Oscar has 25 characters of ciphertext, he will be able to determine a decryption for the ciphertext.